Appendix

A Surrogate Model

This section covers specific implementation details about the surrogate model as well as observations on its performance.

A.1 Implementation Details

Dataset To create training data for the surrogate model $\hat{f}_\phi$, we used the phosphene model described in [24] and implemented in pulse2percept v0.8 [63]. $50,000$ stimuli were created by first selecting a number of electrodes to stimulate between 1 and 30 randomly chosen electrodes, then randomly selecting an amplitude between 1 and 10 (specified as a multiple of the assumed threshold current) and frequency between 1 and 200 Hz for each electrode. In addition, between 10 and 100 electrodes were chosen to act as "noise" electrodes, where either amplitude or frequency was given a nonzero value, but not both. The purpose of these electrodes was for the surrogate model to learn that both a nonzero amplitude and a nonzero frequency are required to produce a visible percept. We used an 80-20 train-test split. As the surrogate model is highly dependent on patient-specific parameters ϕ, we generated new data and fit a separate surrogate for each of the following ϕ: $((\rho, \lambda) \in \{(150, 100), (150, 1500), (800, 100), (800, 1500)\})$.

Network Architecture The surrogate model $\hat{f}_\phi$ used a fully-connected architecture. The input to the model was a stimulus matrix $\mathbf{s} \in \mathbb{R}_{\geq 0}^{n_e \mathrm{x} 3}$, which was identical to the input to f. The stimulus matrix was split into amplitude and frequency components (pulse duration was not used due to poor model performance), which were fed through a FC layer. The outputs of both FC layers were concatenated and fed through another FC layer. Concurrently, the model computed the element-wise product of the amplitude and frequency components and passed it through a separate FC layer. The outputs of the previous two layers were then concatenated and fed through a final FC layer with output size 49×49.

The model was trained for 45 epochs using AdamW [64] optimizer and MAE loss.

A.2 Approximating the Forward Model

The surrogate model was able to accurately approximate the true phosphene model f. Table 2 shows MAE over the validation set ($10,000$ percepts) for all 4 trained $\hat{f}_\phi$. Visually, the predicted percepts were nearly identical to the ground truth.

Table 2: Surrogate model performance

ϕ	$\rho = 150\ \lambda = 100$	$\rho = 150\ \lambda = 1500$	$\rho = 800\ \lambda = 100$	$\rho = 800\ \lambda = 1500$
MAE	0.0119	0.0189	0.0078	0.0115

A.3 Predicted Stimuli

Despite the low surrogate validation error, training with the surrogate model would often result in the encoder suggesting almost adversarial stimuli; that is, stimuli that if fed through the true forward model f would lead to drastically different percepts than if fed through the surrogate model $\hat{f}$ (see Fig. A.1). With these adversarial-like stimuli, the encoder appears to be performing well under the surrogate model, but performs poorly when the same stimuli are input to the true forward model. We identify this as the primary disadvantage of using a surrogate model and resolving this issue remains an open research problem for end-to-end training with surrogate methods.

We noticed several issues caused by the effects of varying stimulus parameters on phosphene appearance. For example, increasing amplitude increases size and brightness, while increasing frequency increases brightness only. We noticed a larger mismatch between the surrogate and the forward model on the extreme ends of the spectrum (*e.g.* very high frequency, low amplitude),

resulting in the encoder settling into a minimum that does not exist in the true forward model. It is important to note this disparity appears despite a high training accuracy of the surrogate alone. Although these examples are specific to the bionic vision application, we expect surrogate models derived to describe other neuromodulation technologies to suffer from similar limitations.

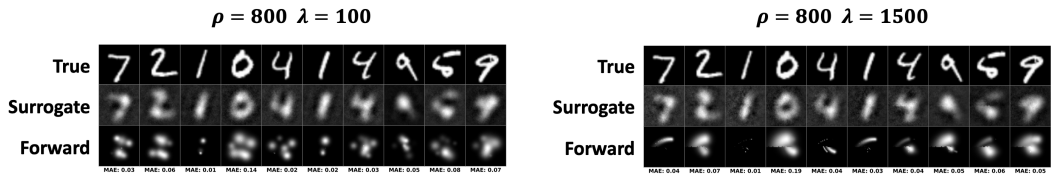

Figure A.1: The encoder would often suggest stimuli that lead to drastically different percepts when fed through the surrogate model ($\hat{f}$, middle row) as compared to the true forward model (f, bottom row). Examples are shown for $\rho = 800$; $\lambda = 100$ (*left*) and $\rho = 800$ $\lambda = 1500$ (*right*).

B Hyperparameter Selection

In this section, we detail how HNA hyperparameters (l, k, α, and β) were chosen.

VGG Loss To choose the layer of the VGG network to use for VGG loss (l) we performed cross validation across a set of candidate layers. Previous studies [52] have shown that the first layer with each of the 5 convolutional blocks perform well for neural style transfer. Thus, we choose these as our candidate layers. For cross validation, we trained HNA for 50 epochs using each candidate layer. The resulting phosphenes are shown in Figure B.1. Using earlier layers, the VGG term performs similarly to MAE, and phosphenes are disconnected. We chose layer 5_1 based on its perceived ability to capture high-level perceptual differences between images, although layer 4_1 also performs similarly.

Figure B.1: Phosphenes produced by HNA encoder with different layers chosen for VGG loss. Layer 5_1 denotes the first layer within the fifth convolutional block.

Laplacian Smoothing We chose to use a kernel size 5 for the Laplacian filter used to estimate the second derivative (k, Eq. 6). The size of the filter controls the scale on which smoothing is applied (*i.e.*, smaller filters sizes only encourage continuity within a small local region, whereas larger filters encourage continuity within a larger region). Size 5 was chosen because larger filters were observed to over-smooth the image, while smaller filters still led to highly disconnected phosphenes.

Joint Perceptual Metric We performed cross validation to find the best values for α and β. Instead of using one value, we found scheduled weighting to be crucial for performance. The scheduler incrementally increased the weight of the VGG loss (β) from 0 while simultaneously decreasing the initially high weight on the smoothing constraint (α). This was motivated by the observation that the VGG loss performed poorly during early iterations when the predicted phosphene was near-random.

Under this scheduled weighting strategy, the loss is dominated early on by the MAE and smoothing terms. This encourages the the model to just output reasonable encodings. As training progresses, the predicted phosphenes become higher quality, causing the VGG loss to perform better, and thus the smoothing term is no longer as important.

Additionally, we found it beneficial to temporarily decrease the learning rate by a factor of 10 for a short 'warm-up' duration following each increase in β, before resetting to 50% of the prior learning rate. This results in the learning rate gradually decreasing throughout training by a factor of around 100. Throughout the paper, we use $\alpha = 0$ and $\beta = 0.00008$ for comparisons of loss values.

C COCO Dataset

For the COCO task (Section 5.2), we used subset of images from the MS-COCO dataset [56]. MS-COCO was chosen due to its selection of common household objects relevant to the daily life of prosthesis users, as well as availability of ground-truth segmentation masks. To select the images suitable for prosthetic vision, we filtered out images according to the following criteria:

1. **Too cluttered.** Any image with greater than 15 total objects was removed. Removed: 15566
2. **Select chosen objects.** Any image that did not have at least 1 object from the selected categories that was larger than 4% of the total image was removed. Removed: 42289
3. **Too many.** Any image with greater than 5 objects meeting criteria 2 was removed. Removed: 1017
4. **Too dim.** Any objects in the image with average pixel brightness less than 50 were discarded. If this resulted in an image having 0 remaining objects, the image was removed. Removed: 434

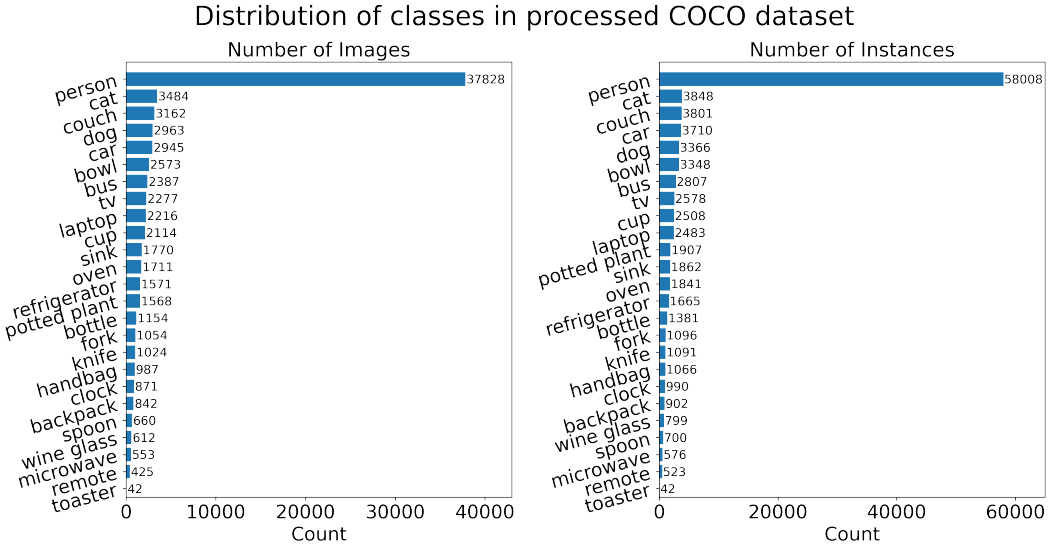

Figure C.1: Number of images (*left*) or instances (*right*) of each category in the processed COCO dataset.

This resulted in a total of 47,532 training images and 11,883 test images (80-20 train-test split). The objects in the remaining images were segmented out using the ground-truth segmentation masks, resized to (49, 49), and converted to grayscale. The distributions of classes used is shown in Figure C.1.

D Predicted Stimuli

Here, we directly examine the stimuli resulting from HNA and naive encoders. Stimuli and their resulting phosphenes for example images from the test set are shown in Figure D.1. The naive encoder produces stimuli with constant frequency (20 Hz) and pulse duration (0.45 ms), which are not shown.

We make the following observations about the predicted stimuli:

- Both encoders activate electrodes corresponding to the shape of the target image. In naive stimuli, the amplitude directly corresponds to the pixel brightness. In HNA stimuli, the distributions of amplitude, frequency, and pulse duration across the electrodes is more complex and harder to characterize, but lead to higher-quality phosphenes.
- HNA uses amplitudes inversely proportional to ρ.
- For small ρ, HNA primarily uses amplitude to control brightness. For large ρ, HNA primarily uses frequency to modulate brightness, keeping amplitudes low to limit phosphene size.
- HNA uses small pulse durations to create lines parallel to the underlying axon NFB (*i.e.*, it utilizes the streaked phosphenes to its advantage), and large pulse durations to create lines perpendicular to the underlying NFB. In other words, HNA was able to exploit application-specific (*i.e.*, neuroanatomical) information that is baked into the forward model.
- On average, HNA uses more electrodes, larger frequencies and pulse durations, and smaller amplitudes than the naive encoder. A large active electrode count and high pulse durations may not be desirable for some prostheses, due to tissue activation and frame rate limits. We found that it was easy to constrain these parameters using regularization on the output stimuli, at the cost of slightly decreased performance.

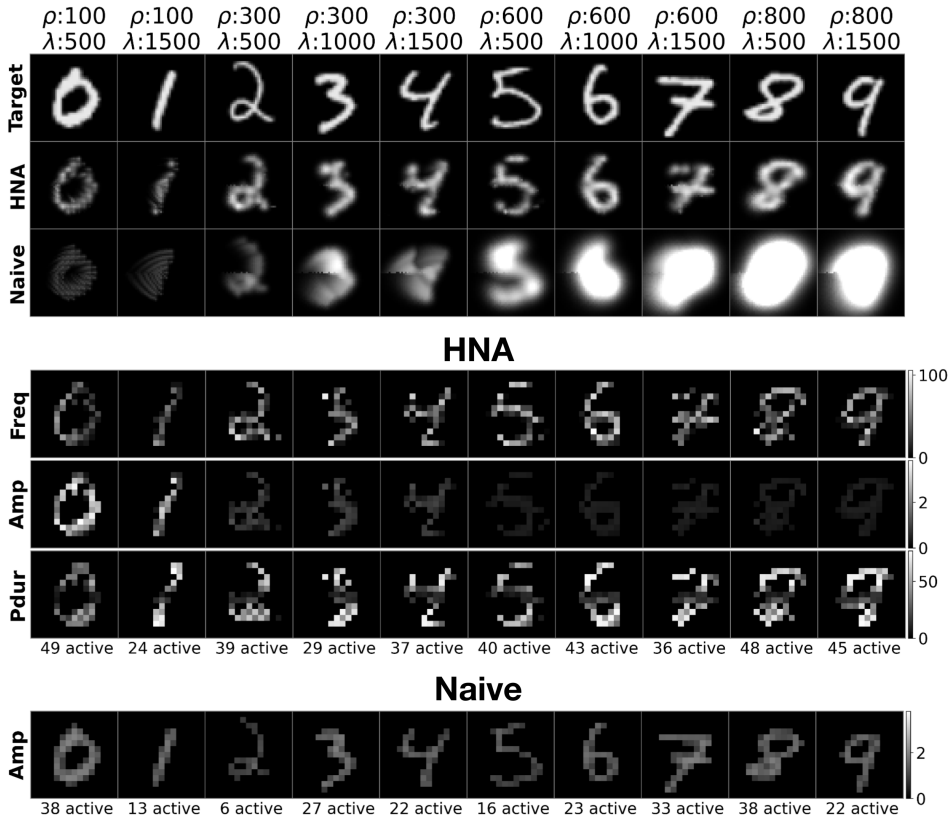

Figure D.1: *Top*: Example MNIST target images, and the phosphenes produced by HNA and naive encoders, encoded at various ρ and λ values. *Center*: The stimuli corresponding to the HNA phosphenes. From top to bottom, stimulus frequency (Hz), amplitude (xTh), and pulse duration (ms) are shown. The number of 'active' electrodes stimulated above threshold levels is given below each stimuli. *Bottom*: Stimuli corresponding to the naive phosphenes.

E COCO Patient-to-Patient Variations

We repeated the analysis presented in Section 5.3 for the COCO dataset. Figure E.1A shows two example COCO images, encoded by both HNA and the naive encoder, across varying ρ and λ values. The heatmaps in Figure E.1B show the log of the joint perceptual loss across simulated patients, for both the naive and HNA encoders. To measure phosphene consistency, we performed T-SNE clustering on a subset of the COCO images which have only 1 object. Unfortunately, T-SNE clustering of the ground-truth COCO images did not form groups corresponding to the object types (Figure E.1C), suggesting that the representation of object instances vary drastically across COCO images. Therefore, it was not meaningful to repeat the analysis presented in Fig. 5C.

Similar to the MNIST results presented in Section 5.3, HNA produced higher-quality representations than the naive encoder, resulting in a lower joint loss for every simulated patient. HNA performed consistently well across all simulated patients (Figure E.1B), with a small increase in loss for small ρ (< 100). Similar to MNIST, the naive encoder only performs well for patients with a mid-to-low ρ (≈ 200) and low λ.

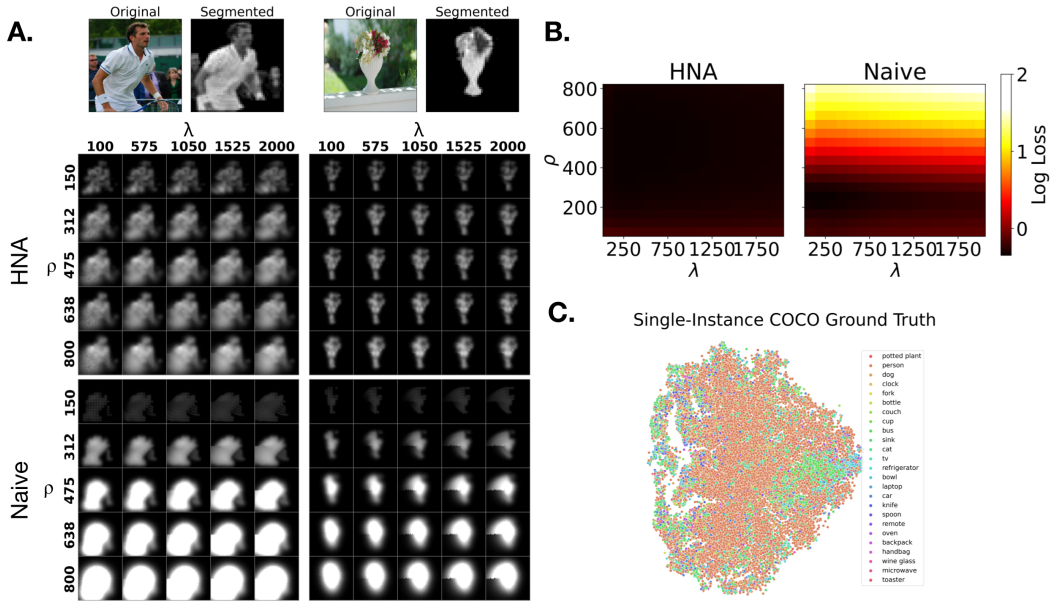

Figure E.1: COCO Encoder performance across simulated patients (varying ρ and λ). **A:** Phosphenes produced by HNA and Naive encoders of two example images. **B:** Heatmaps showing the log joint loss across ρ and λ for HNA and naive encoders. **C.** Ground-truth COCO images cannot be clustered using T-SNE into groups corresponding to the object types. The clustering was performed on COCO images that only contained one object.

F Modeling Other Patient-to-Patient Variations

Previously, results were presented across patient-specific parameters ρ and λ, because these have the greatest impact on phosphene appearance. However, the forward model has a number of other patient-specific parameters, which HNA is also able to adapt to. For full details on all parameters of the forward model, see [24]. Out of the remaining parameters, $a2$, $a3$, and $a5$ are the most impactful on phosphene appearance. $a2$ and $a3$ modulate how much the brightness contribution from each electrode scales with increasing amplitude and frequency, respectively. $a5$ locally scales the global radial current spread ρ based on each electrodes amplitude. Figure F.1 (*left*) illustrates the effect of these parameters on phosphene appearance.

Figure F.1 compares HNA to naive encoder performance across $a2$, $a3$, and $a5$. The ranges for these parameters are based on values empirically observed in retinal prosthesis users [24]. HNA produces relatively consistent phosphenes, and outperforms the naive encoder across all conditions.

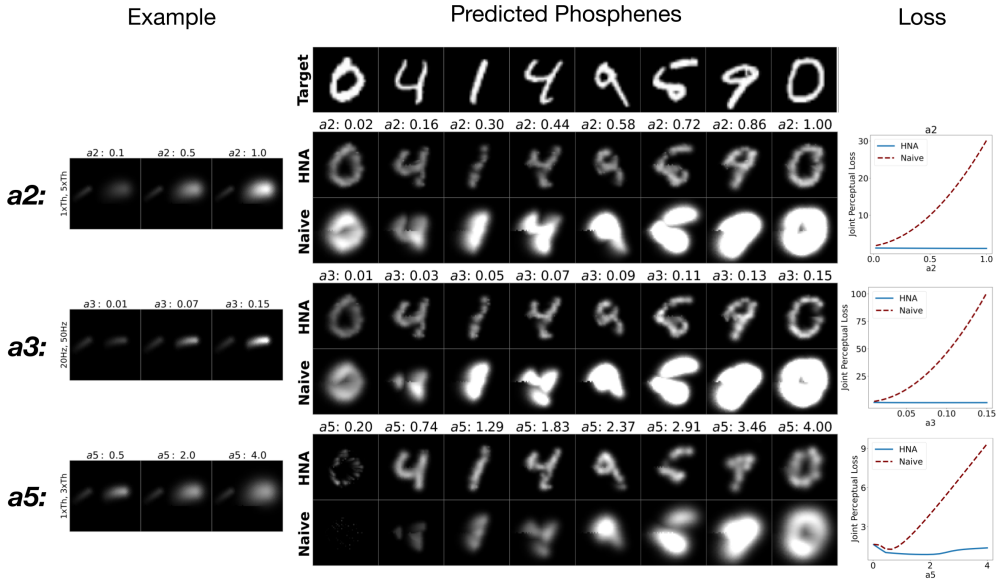

Figure F.1: *Left*: Examples of how a2, a3, and a5 affect single-electrode phosphenes. a2 modulates local brightness scaling with increasing amplitude, a3 modulates local brightness scaling with increasing frequency, and a5 modulates local size scaling with increasing amplitude. *Center*: Phosphenes predicted with HNA and naive encoders for varying a2, a3, and a5, increasing left to right. *Right*: Plot showing the joint loss across a2, a3, and a5 for HNA (solid) and naive encoder (dashed line).

G Simulating Higher-Resolution Implants

On the COCO task, HNA significantly outperformed the naive encoder, but was still unable to capture all of the detail in the images. Two of the main reasons for this are the limited spatial resolution of the implant and the patient-specific distortions from the forward model. Here, we present results from HNAs trained on implants of higher resolution, at small ρ and λ. The chosen implants are illustrated in Figure G.1A. For a fair comparison, each HNA was trained for only 50 epochs.

Phosphenes resulting from the HNA trained on the different implants are shown in Figure G.1C, and the losses across implants is plotted in Figure G.1B. As implant resolution increases, the phosphenes look increasingly similar to the ground truth, and small details (e.g. facial details, textures) start to emerge.

Thus, HNAs initial failure to capture high-frequency details in the image appears to be an application-specific limitation for visual prostheses more so than a limitation of the HNA framework. For visual prostheses, learning to reconstruct the high-frequency features of complex images despite distortions and limited implant resolution remains an open problem.

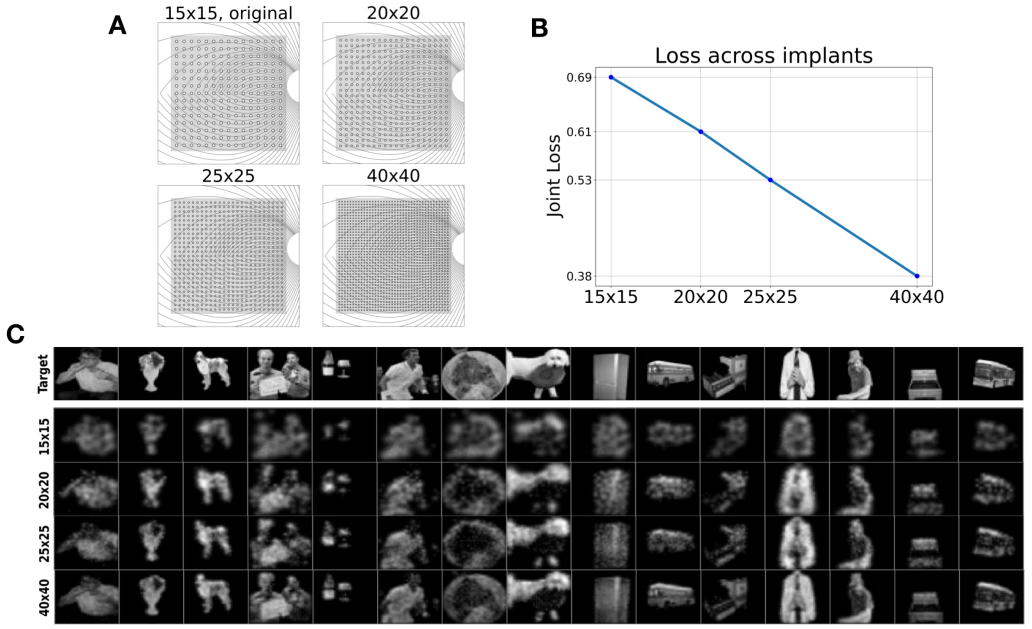

Figure G.1: **A**: The 4 different implants compared. The main text uses the 15×15 implant. **B**: The joint perceptual loss of HNAs trained on the different implants after 50 epochs. **C**: Example images showing the reconstructed phosphenes using each implant

H Mis-Specified Patient-Specific Parameters

Due to noisy or limited patient data, there may be some uncertainty in the measured value of the patient-specific parameters ϕ. Therefore, we conducted an analysis of the consequences of incorrect patient-specific parameters on the encodings produced by HNA. Note that the true patient-specific parameters are not needed during training, so incorrect ϕ will only affect evaluation. A 'mismatch' HNA model was created, where the forward model decoder used the true patient-specific parameters ϕ, and the encoder used another set of patient-specific parameters ϕ'.

In the first experiment, ϕ' was sampled from a uniform random distribution (we again focus on only ρ and λ). The original HNA encoder, naive encoder, and mismatch HNA encoder with random ϕ' were evaluated on the MNIST test set. HNA achieved a joint loss of 0.92, the naive encoder had a joint loss of 3.13, and the mismatch HNA had a joint loss of 1.35 ± 0.003 (mean $\pm$ standard deviation across 10 random ϕ'). Thus, even if the true patient-specific parameters are completely unknown, on average randomly selecting values will still produce higher-quality encodings than the naive method.

In a second experiment, we analyzed whether there were any configurations (ϕ - ϕ' combinations) that resulted in a worse encoding than the naive model. For the 90% of true ϕ, the mismatch model outperformed the naive model regardless of the chosen ϕ'. However, the naive model performs best at $\rho = 250$ and $\lambda = 200$. In Figure H.1A, we hold λ constant at 200 and, for each true ρ, plot the ranges of mis-specified ρ' for which the mismatch HNA still outperforms the naive. Figure H.1B shows a similar plot for varying λ, holding ρ constant at 250. Even for the naive model's ideal patients, HNA still outperforms the naive model for a large proportion of mis-specified ρ and λ.

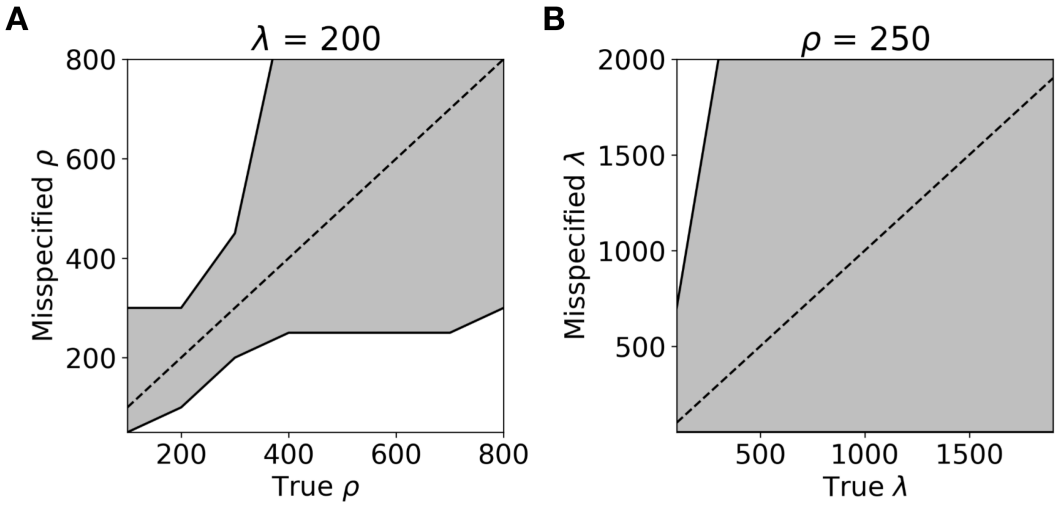

Figure H.1: Plots showing mis-specified HNA performance relative to the naive encoder for varying ρ (panel **A**) and λ (panel **B**). The dashed line marks the correctly specified model, and shaded area between the solid lines shows the region where the mis-specified HNA outperforms the naive encoder. Note that the naive model's ideal patient was used, with λ fixed at 200 and ρ fixed at 250, respectively.